



Towards a graphical and real-time network simulation toolset

Sterritt, R., Curran, E., & Adamson, K. (1998). Towards a graphical and real-time network simulation toolset. In R. A. Adey, G. Rzevski, & P. Nolan (Eds.), *Applications of Artificial Intelligence in Engineering XIII* Computational Mechanics Publications. <https://www.witpress.com/elibrary/wit-transactions-on-information-and-communication-technologies/20/13880>

[Link to publication record in Ulster University Research Portal](#)

Published in:

Applications of Artificial Intelligence in Engineering XIII

Publication Status:

Published (in print/issue): 01/07/1998

General rights

Copyright for the publications made accessible via Ulster University's Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk.

Towards a graphical and real-time network simulation toolset

R. Sterritt, E.P. Curran, K. Adamson

School of Information and Software Engineering,

Faculty of Informatics, University of Ulster,

Newtownabbey, Co. Antrim, Northern Ireland.

EMail: {R.Sterritt, EP.Curran, K.Adamson}@ulst.ac.uk

Abstract

Using simulation to model complex engineering systems can provide cost-savings for testing, while being able to stretch boundaries not normally feasible on actual system tests. The authors have been involved in researching and applying a toolset to provide such testing, in terms of behaviour analysis, for SDH Multiplexer telecommunications systems. This actual simulation has been implemented on transputers to provide scaleable and flexible topology emulation and a KBS tool has been prototyped to perform intelligent mapping between the desired SDH topology and transputer simulation.

1 Introduction

In Industry, simulation is a cost-effective and rapid method for testing and designing new equipment and systems. At NITEC (Northern Ireland Telecommunications Engineering Centre), a part of BNR Europe and the R&D site for NORTEL, they design and manufacture SDH multiplexers(CCITT¹). They carry out testing of the new components on their internal SDH test network which typically consists of six multiplexers (Figure 1).

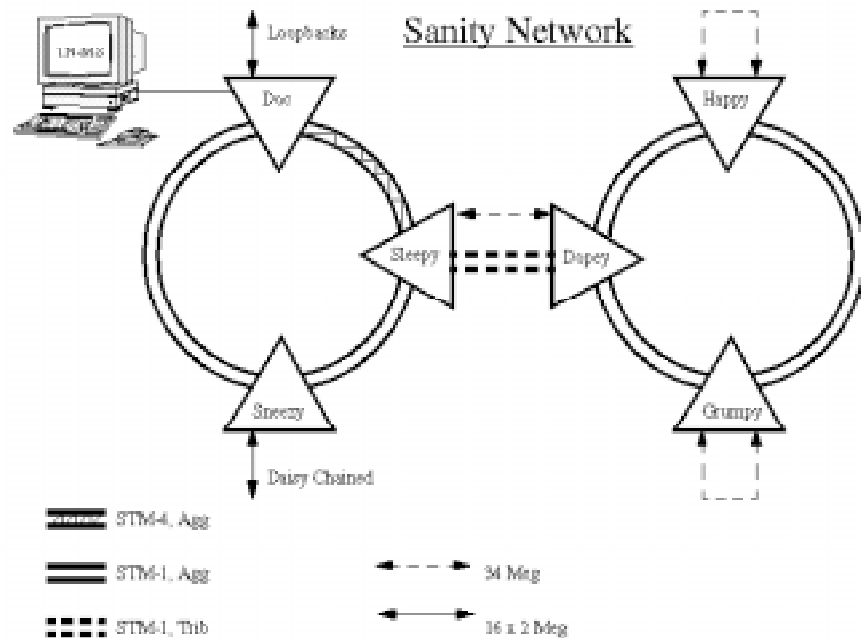


Figure 1 NITEC SDH Test Network

Although this is satisfactory for ensuring correct function and integration of new components into the multiplexer it does not give them satisfactory testing facilities for different topologies and large scale network behaviour.

Due to the expense of these multiplexers and the space requirement to build a large network for testing purposes, network testing happens in the field once the network is implemented (Sterritt et al²).

2 The Toolset

Nortel have identified a need for a tool which will emulate the behaviour of a network of SDH multiplexers. To be useful in large system testing, the tool will need to be capable of modelling a wide range of SDH multiplexer types and topologies and should allow real components (e.g. the STM-1 multiplexer) to be connected into the emulated network for test purposes.

Several graphics-based network simulation software products already exist. These were investigated to assess their suitability. This process was generally carried out by installing the products on site in NITEC and subjecting them to in-depth testing. Products investigated in this way included BONEs, NETWORK II.5, COMNET, and OPNET.

The results of this investigation made clear that none of the products assessed would be suitable for use as the network emulation tool. Different products exhibited varying strengths and weaknesses but one criterion was found to be critical as no product was capable of performing sufficiently to satisfy the requirements of real-time implementations.

This outcome is hardly surprising as all the packages investigated are implemented in software only and are not intended to perform at real-time hardware speeds. This requirement is, however, crucial to the proposed network emulation tool. If a hardware component under test is to be connected directly to an emulated network, the emulated network must, consequently, be performing at real-time hardware speeds. BONEs 2.0 was the product which came closest to matching the performance requirements, but even when modelling small networks of incomplete multiplexers (six nodes), performance figures were found to be between 16,000 and 20,000 times greater than real-time.

In all test cases simulations were stripped to the bare essentials required for execution, with, for example, no network animation on-screen.

Since no tool suitable for use as the network emulator was readily available, academic research into how a tool might be developed was carried out jointly between NORTEL and the University of Ulster at Jordanstown.

The investigation concluded that the method most likely to yield significant results would be to use arrays of transputers (INMOS³) to increase the processing power of the emulation tool. As transputers can be acquired in arrays of, for example, sixteen units interconnected on a single motherboard and motherboards can be connected together in racks, the only theoretical limit on the number of networked multiplexers emulated would be the number of transputers available to emulate the network elements in a supplied network topology.

Other researchers have also come to this conclusion and implemented simulations using multiprocessors; Bhimani et al⁴, Chai et al⁵, Nichols et al⁶, Manthorpe et al⁷ and Richardson et al⁸.

As a result a toolset was designed with the aim of developing a workstation and transputer based Graphical And Real-time Network Emulation Tool (GARNET).

This potentially will allow many companies who manufacture, design or purchase SDH/B-ISDN network components to perform cost-effective large scale system testing of these components for the first time. The size, complexity and financial investment involved in SDH/B-ISDN wide-area

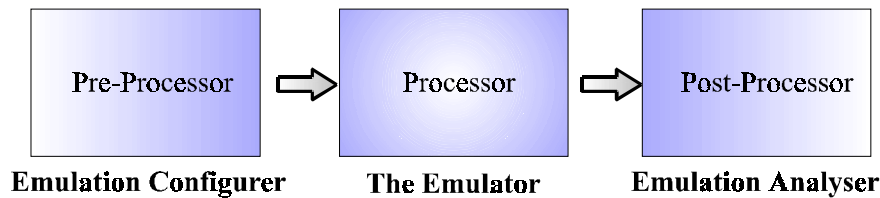


Figure 2 GARNET High Level Components

networks has meant that large scale system testing of such hardware was only possible for companies able to allocate the considerable amount of resources required to perform this task. For such companies, the emulation tool would enable this operation to be carried out more successfully by cutting costs, scaling down the test environment and allowing an overall “de-skilling” of the process.

The design has three high-level modules (Figure 2); first the *Emulation Configurer*, second the *Emulator* (main sub component being the SNE - Synchronous Network Emulator) and third the *Emulation Analyser*.

2.1 The Emulation Configurer

In the GARNET project, the use of transputers that require configuration for each type of topology implemented added an extra level of complexity. If the objective of “de-skilling” the process of large system testing of network components was to be met, a graphical tool was essential (Sterritt et al.⁹).

To ensure the topology was mapped onto the transputers providing a solution as near to real-time speeds as possible requires intelligent configuration of the transputer network, which must be incorporated into the graphical tool.

2.1.1 TopMode (Topology Mode)

TopMode is designed to enable the user to draw graphically the required SDH/SONET Network topology to be emulated.

It will present the user with a window containing a canvas where the desired topology will be drawn and a panel of buttons which will allow the user to “draw” the topology (Figure 3). The user will also be able to save their topology once it is drawn so that common topologies do not have to be redrawn every time the emulation is run.

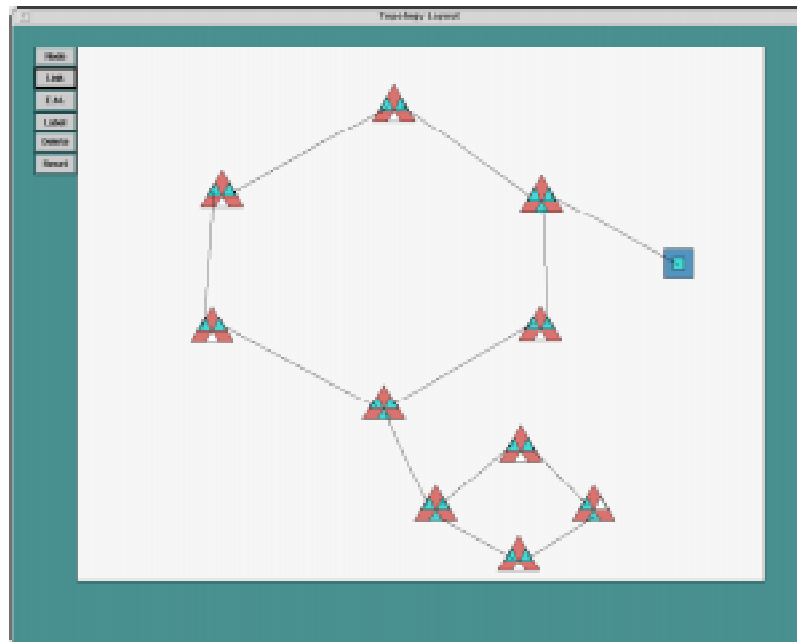


Figure 3 Screenshot of TopMode⁹

The advantage in separating the function into sub-components is that it de-couples the function. It is possible in the future to design and implement TopMode in another system that will still work with RecMode where the intelligence resides. This would allow the development of a geometrical visualisation interface which can potentially allow easier visualisation of hundreds of nodes.

2.1.2 RecMode (Reconciliation Mode)

RecMode will tie together the information supplied in TopMode and HardMode so that the transputer network can be configured to emulate the topology specified in TopMode. The RecMode window (Figure 4) contains three buttons: check, configure and run.

The “Check” button will count the number of nodes and element managers (EM) in the topology and the number of transputers specified in HardMode and will check that there are enough transputers to represent the topology. In the case of too few transputers an error message will appear on the screen as soon as the “Check” button is pressed. The check button also will verify that every slot on the transputer board is occupied either by a transputer or a jumper. If this is not the case an error message will be posted.

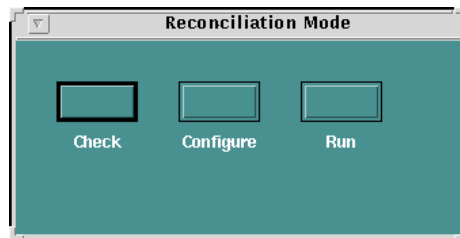


Figure 4 Screenshot of RecMode

The second button, “Configure”, will produce a configuration file (Figure 5) which can be used to configure the transputer network to represent the topology specified in TopMode.

The final button, “Run”, will activate the configuration process by sending the configuration file to the transputers, will set up the virtual network and will execute the appropriate code on each of the transputers.

```
/* **** */
/* GARNET */
/* Configuration file for SDH network model */
/* **** */
/* Define the processors */
T800 em;
T800 gateway;
T800 mux1;
T800 mux2;
T800 mux3;
/* Reserved memory for user-defined allocation */
em (reserved = 80K);
gateway (reserved = 80K);
mux1 (reserved = 80K);
mux2 (reserved = 80K);
mux3 (reserved = 80K);
....
....
```

Figure 5 Sample Configuration File

2.1.3 HardMode (Hardware Mode)

The purpose of HardMode will be to allow the user to specify what hardware is available for the emulation.

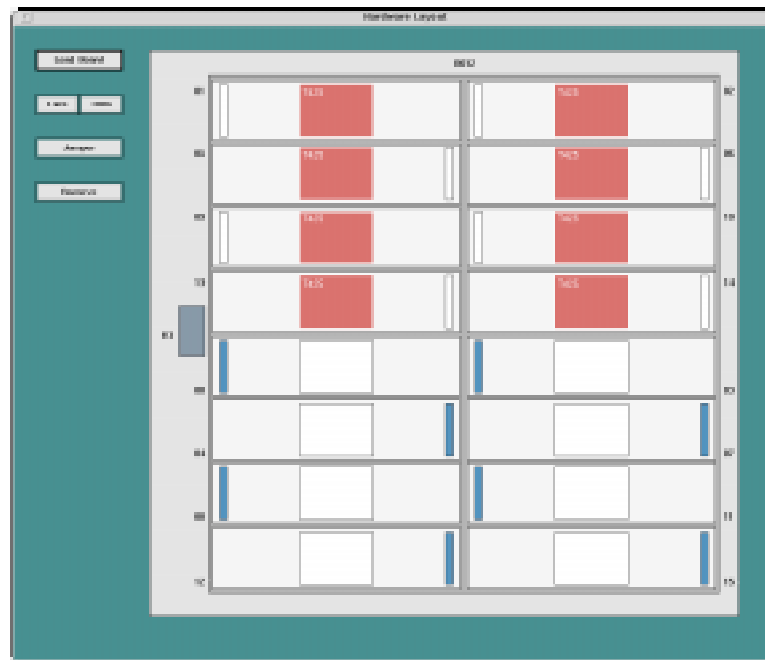


Figure 6 Screenshot of HardMode⁹

Two designs of HardMode were considered. An explicit version where the user or NORTEL engineer would specify the hardware available, and a dynamic version where the specification of hardware would be automated and dynamic.

The dynamic prototype would take advantage of the Network Analyser (rsby), available with the new version of the transputer toolkit from INMOS¹⁰, to dynamically create a specification of the hardware and part of the configuration file.

For the explicit prototype the HardMode window will show a transputer board layout and will allow the user to place different transputers in the appropriate slots and “jumpers” in the empty slots (Figure 6).

The Dynamic approach (Figure 7), where the hardware is dynamically detected by a network analyser, will provide ;a more reliable specification of standard hardware by removing the possibility of human error; remove the need to understand the complexity of the transputer equipment; a method that would automatically and quickly detect any changes in hardware configuration.

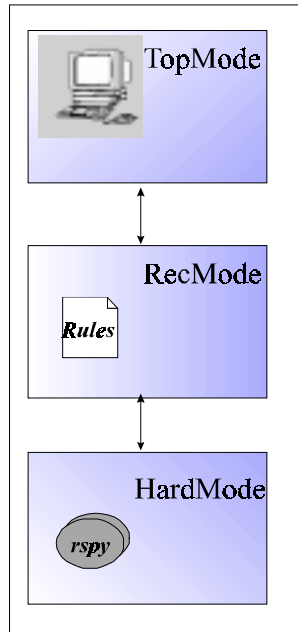


Figure 7 Dynamic Emulation Configurer Layout

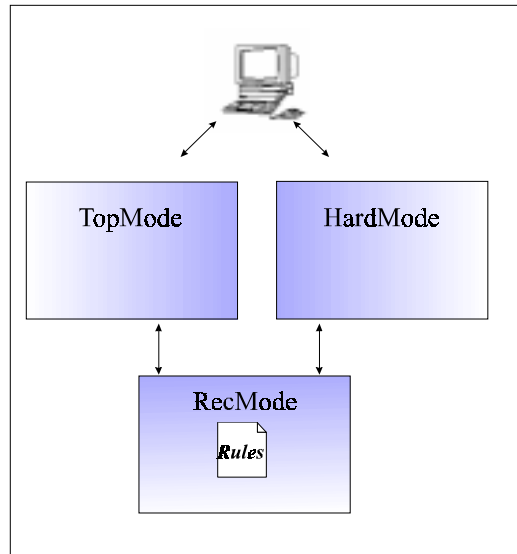


Figure 8 Explicit Emulation Configurer Layout

The Explicit approach (Figure 8), where the hardware will be explicitly defined by the user, will : encourage the NORTEL engineer to get to know how the hardware works and the relationship between it and the emulation; provide a more secure feeling about the application (remove the inherent distrust in automation that has been highlighted by discussion with NORTEL); provide a method to describe the position of bespoke hardware that may be designed by NORTEL (e.g. external clock source).

It was envisaged that the implemented version would consist of a mixture of the two approaches described here, mainly relying on the dynamic approach with the explicit approach being used for unusual circumstances.

2.2 The Emulation Component

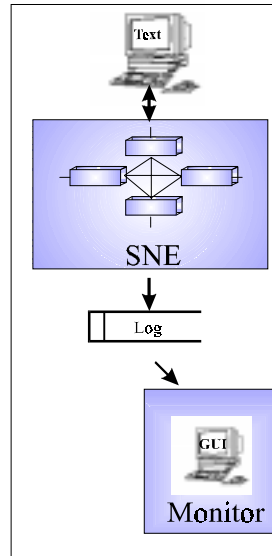


Figure 9 Emulation Components¹³

2.2.1 Synchronous Network Emulator (SNE)

The specific purpose of the SNE (Synchronous Network Emulator) is to model the behaviour of the messages and signals, in particular the alarm and performance signals in the event of faults for large scale tests. The main reason behind the use of transputers in this implementation is the facility for flexible network topologies and its perceived scalability.

The first version of the SNE models three main modules; (1) a simple element manager (EM); (2) a gateway mux and (3) a non-gateway mux. The specification for the multiplexer (mux) is based on CCITT¹¹ requirements with BNR detailed requirements in places where CCITT is not sufficiently specific. The TN-1X is NORTEL's¹² implementation of an STM-1 Add-drop configuration with 2 STM-1 ports and 63 2Mbit/s tributaries.

The SNE models the following behaviour; (1) SDH frame content and transmission; (2) Response to incoming frame content; (3) Response to fault conditions; (4) Management logic and communications and (5) Interface to the outside world at the management interface and SDH frame interface.

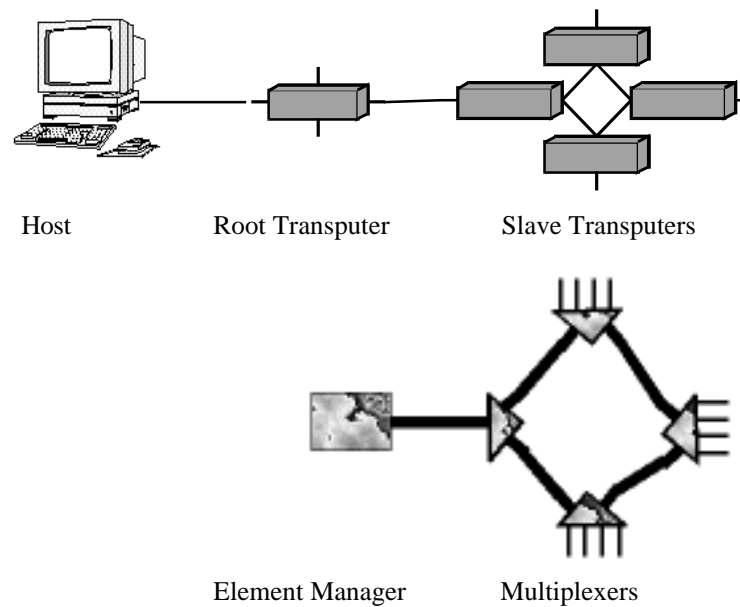


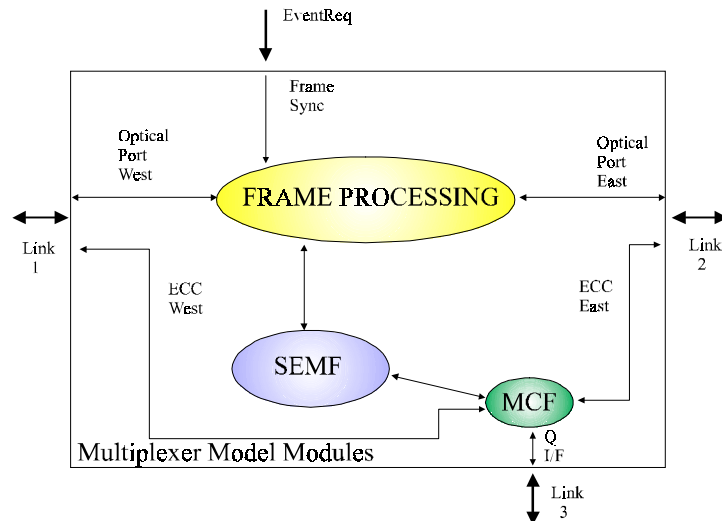
Figure 10 Mapping between SDH Topology and Transputer simulation

The SNE can emulate the network behaviour of a STM-1 (NORTEL product name; TN-1X) multiplexer or a network element manager on a single transputer. For more complex components several transputers may be required to achieve real-time. This adds to the complexity of configuration.

Initial results with the emulator yields near real-time speed, showing that this hybrid approach of a hardware and software solution far outstretches a purely software solution.

The beauty of a transputer-based configuration in this case is the scalability. To increase the size of a SDH network topology emulation merely requires the addition of more transputer nodes. In the STM-1 (TN-1X) and network element manager case where there is a 1:1 mapping (Figure 10), the limit on the topology size is the number of transputers in the system.

The main building blocks of the model program and the use of the transputer links, are shown in Figure 11. The gateway mux, which has the Q interface, is shown.



SEMF : Synchronous Equipment Management Function
MCF : Message Communication Function
ECC : Embedded Control Channel
Q : Interface to External Element Manager

Figure 11 Mux implementation on a Transputer

The box represents a single transputer. Within the box are the software modules of the multiplexer model, with their connections to each other and the external ports indicated (three communication ports and the event port).

SEMF consists of one process with the main functions (alarm handling, message handling, agent and VNET, Shelf/Card handling, & performance monitoring) being cycled around within the process. The MCF transmits asynchronous ECC messages intact. The Frame Processing module controls I/O of the modelled frames and the logic to produce the correct frame rate.

Figure 12 and Figure 13 display the text based interactive user interface to the emulator. This is produced by the root transputer running the Element Manger (EM) process.

Figure 12 displays the main menu which shows the functions possible: Enter faults, connections, masking, cards, Display status, connections, time, performance, equip shelf.

Figure 13 shows option 2, the four possible fault types are displayed in a menu; a card failure has been chosen.

Main Transputer Network Emulator Menu.

1. Start Network
2. Enter a fault
3. Display alarm status of network nodes.
4. Make a Connection.
5. Display Connections.
6. Display Time.
7. Alarm Mask Control (Default - Mask ON).
8. Bit Errors and Performance Monitoring.
9. Debug Messages Transmission: ON/OFF.
10. Equip Shelf and Add/Remove Cards.
11. End.

ENTER USER SELECTION (1..11):

Figure 12 Screenshot of SNE Menu

Enter Fault Type: 0 = High-Order
1 = High-Order Channel
2 = Low-Order
3 = Card

Selection: 3

Enter Card Fault 0 = Card Fault
1 = Card Fail

Selection: 0

Enter slot number of card : 2

Enter Duration of Fault in milliseconds
(-1 = Indefinite):

Selection: 10000

Figure 13 Screenshot of Option 2

2.2.2 Monitor

When implementing the simulation there was a conflict in the desire to build a totally graphical interface and the near real-time requirement. Due to the extremely high rate of network message generation it was thought to be unrealistic to show a network topology graphically and network messages “animated” in real-time as they pass from node to node.

Due to the speed issue the first version of the simulator was implemented with a text based interface. Simulated network monitoring and the entry of faults to the simulator are via a textual menu system and the events and alarms from the network are reported back via the root transputer (which simulates a SDH Network Manager) to a log file on the UNIX host for off line analysis. The user can view this log file during usage of the simulation by using the UNIX 'tail' command. The reports in the log are quite cryptic to the non-telecommunications engineer.

The solution developed by the team was a non-invasive GUI, thus the term Monitor, to report the behaviour in a more meaningful and user friendly way. This approach does not infringe on the requirements of near real-time and provides an additional optional tool (Gilmour¹³).

The virtual node display screen shows, in the top section, each of the mapped SDH multiplexers to transputers. A right-mouse button click on the chosen node provides a pop-up menu with a choice of Node, Transputer and Alarm which will then display information about the chosen option that relates to that node in the bottom part of the screen.

The imagebar to the left offers different screens. The buttons are greyed-out until the mouse pointer is above it. The options are Node Display (Figure 14), View Configuration (Figure 15), and View Alarm (Figure 16). The imagebar stays resident when a different screen is displayed.

The configuration file display screen shows the transputer configuration used to determine the set-up of the emulator. This file would have been produced by the Emulation Configurer (EC), and as such the information to determine what the SDH topology is may be available to the system.

It was decided the Monitor should obtain the topology from the configuration file and not from the EC in order to uncouple it from the EC. The EC is not necessarily used every time the Emulator is used, and it is quite feasible that the configuration file could be modified or be written by the user.

The alarm file display screen shows the alarm log. It is this log that is produced by the Emulator (SNE). The monitor takes its information about the current status of the Emulator from this log fulfilling the non-invasive objective.

The Monitor prototype has met the requirement of providing a non-invasive GUI for the Emulator. Yet the main aim of this additional tool was to make the output from the SNE less cryptic to the non telecoms engineer. To a great extent this has not been a success due to Java. It was imagined that the SDH Topology could be simply displayed giving an immediate visualisation of what is being emulated (like in the Emulation Configurer,

which was implemented in PowerModel¹⁴). The behaviour of the SNE would then be traced by such means as flashing SDX (multiplexers) on this topology. The problem was that Java does not like absolute positioning, in its aim for platform portability. The result was that the SDXs are just placed beside each other. However, Java did allow a very neat OO implementation of the inner workings.



Figure 14 Screenshot of the Node Display¹³



Figure 15 Screenshot of the Configuration Display



Figure 16 Screenshot of the Alarm Display

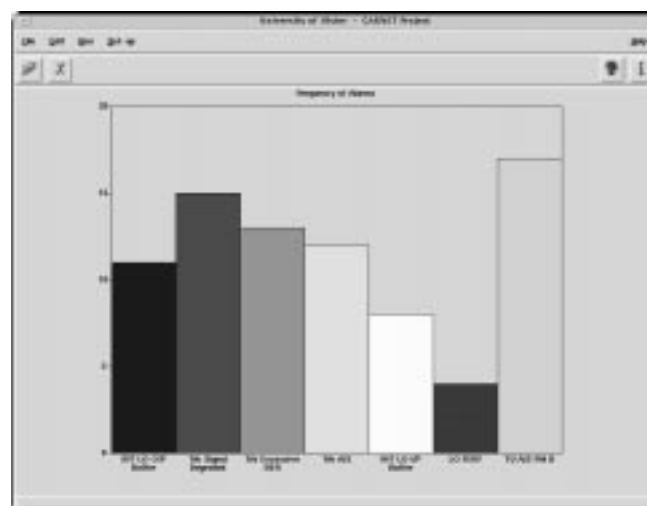


Figure 17 Screenshot of Emulation Analyser's Bar Chart Tool

2.3 The Emulation Analyser

The Emulation analyser provides a means of post analysis of the alarm data produced in the emulation. The prototype reads the log file and produces

statistics on alarms raised which are used to create a bar chart showing the frequency of different types of alarms (Figure 17). This data summarises the behaviour of the network during the emulation and can be used by the users to determine how it differs from the expected behaviour of the network and enable refinement.

3. Conclusion

The full prototype toolset is shown in Figure 18. The Emulation Configurer was implemented in PowerModel (C & ProTalk¹⁵), the Emulation Component in INMOS C (Parallel C) and Java (the Monitor) and the Emulation Analyser in TCL/TK and GNU C.

The GARNET project, so far, has produced a viable design and prototype to address the research and industrial partner's goals. It now stands at the stage where the design and prototype can be built on to produce a full implementation.

Since one of the high level objectives of the project is to provide a means to facilitate large-scale flexible (multi-topology) SDH simulation testing the

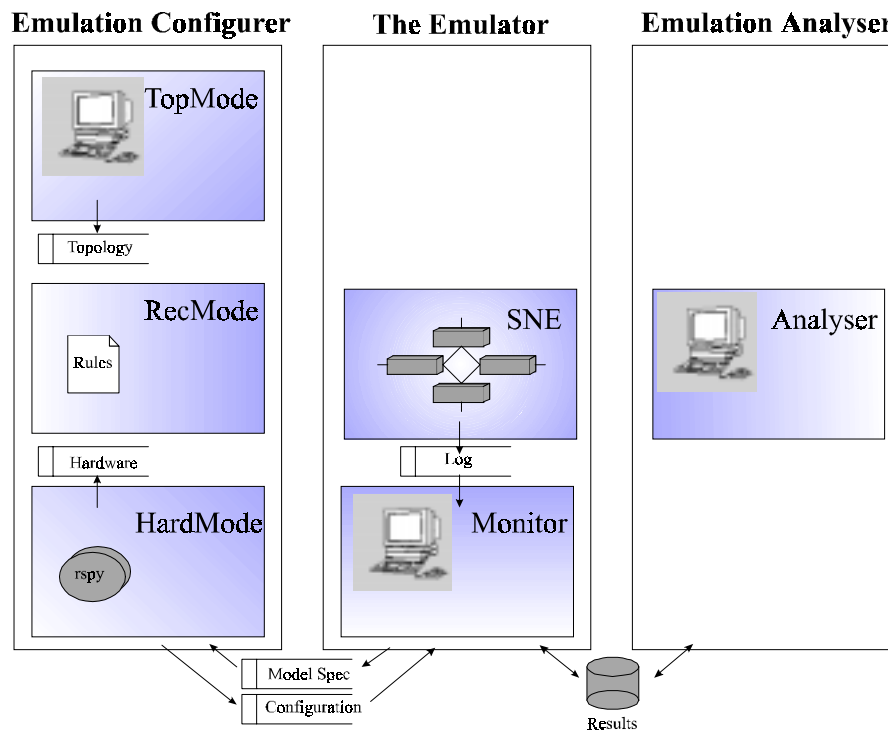


Figure 18 GARNET Toolset Detail

following further developments would benefit the work:

1. To allow for large scale networks with potentially hundreds of nodes, TopMode could be redesigned represent hundreds of nodes better by using geometrical visualisation techniques(Leung et al¹⁶).
2. The SNE has been developed to STM-1 standard, to provide fuller function this would need to be expanded to STM-4 and STM-16.
3. The Emulation Analyser could utilise AI techniques to prove and refine the SNE simulation.

Acknowledgments

We would like to thank IRTU for funding this work and our industrial collaborators BNR NITEC (R&D facility for NORTEL), and Transtech Parallel Systems Ltd. for their support of this work.

References

- [1] CCITT, *Types and General Characteristics of SDH Multiplexing Equipment*, CCITT Recommendation G.782, 1990.
- [2] Sterritt, R., Livingstone, N., Curran, E., Adamson, K., A Graphical And Realtime Network Emulation Tool, *Poster Displayed at the EPSRC Architectures for Integrated Knowledge-Manipulation Systems 3rd Annual Conference*, 1997
- [3] INMOS Limited, *Transputer Development System*, Prentice Hall, ISBN 0-13-928995-X, 1988
- [4] Bhimani A., Ghosh S., Modeling and Distributed Simulation of Complex Broadband ISDN Networks under Overload on Loosely-Coupled Parallel Processors , *Proc. Int'l Conf. Comm.*, IEEE Comm. Society, Piscataway, N.J., pp 1280-1284, 1992.
- [5] Chai A., S. Ghosh S., Modelling and Distributed Simulation of a Broadband-ISDN Network ,*Computer* Vol 26 No 9, Sept 1993.
- [6] Nichols S.J., Clarke R.T., Mars P., Design of a High Speed Simulation Tool for WAN using Parallel Processing, *Microprocessing and Microprogramming* - Vol 25 pts 1-5 pgs 327-332, Jan 1989.

- [7] Manthorpe S., Phillips C., Cuthbert, L.G., High Performance ATM Network Simulation using Transputers, *Colloq. on Parallel Processing*, (C13), Digest No 1990/122, Jun 1990.
- [8] Richardson D., Day B., Using Parallel Processing for Simulation, *ESC 89, Proc. 3rd. European Simulation Congress*, Sep 1989.
- [9] Sterritt R., Curran E.P., Adamson K., Houston P., Towards An Intelligent GUI for an SDH Simulation on a Transputer Network, *Proc. 16th IASTED Int. Conf. Applied Informatics*, 1998
- [10] INQUEST *User and Reference Manual*, SGS-Thomson Microelectronics, (72 TDS 405 05) Oct. 1995
- [11] CCITT, *SDH Management*, CCITT Recommendation G.784, 1990.
- [12] FiberWorld *TN-IX Product Handbook*, Northern Telecom, 1994
- [13] Gilmour G., Sterritt R., Adamson K., Curran E.P., A Monitor For An SDH Simulation On A Transputer Network, *Proc. 16th IASTED Int. Conf. Applied Informatics*, 1998
- [14] IntelliCorp, *PowerModelTM User's Guide*, IntelliCorp Inc. (PM3.2-UG-1), Mar. 1996
- [15] IntelliCorp, *PowerModelTM ProTalk Language Reference*, IntelliCorp Inc. (PM3.2-PT-1), Mar. 1996
- [16] Leung Y.K., Apperley M.D., A Review and Taxonomy of Distortion-Oriented Presentation Techniques, *ACM Trans. Computer-Human Interaction*, Vol. 1 No. 2, June 1994 pp126-160